**Developing a chatbot using AWS Lex and Lambda**

*A Project Based Learning Report Submitted in partial fulfilment of the requirements for*

*the award of the degree*

*of*

**Bachelor of Technology**

**in The Department of Electronics and Communication Engineering**

**CLOUD & SERVERLESS COMPUTING**

**(22CEC3305A)**

Submitted by

**2210040139: Pranathi**

**2210040040: Anusha**

**2210040122: Shiva Prasad Reddy**

**22100400118: Mahalakshmi**

*Under the guidance of*

**Ms. Saritha**

**Course Coordinator**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**K L Deemed to be UNIVERSITY**

*Aziz Nagar, Moinabad, Hyderabad,*

*Telangana, Pincode: 500075*

2024-2025

**DECLARATION**

We hereby declare that the project entitled “Developing a chatbot using AWS Lex and Lambda” which is being submitted as project-based learning of 6th semester in Electronics and Communications Engineering, Aziz Nagar, Hyderabad in authentic record of genuine work done under the guidance of Assistant Professor Ms. Saritha, Department of Computer Science and Engineering, KL University, Aziz Nagar, Hyderabad.

Date:

Place: Hyderabad

|  |  |
| --- | --- |
| Pranathi | 2210040139 |
| Anusha | 2210040040 |
| Shiva Prasad Reddy | 2210040122 |
| Mahalakshmi | 2210040118 |

**K L Deemed to be UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

*Certificate*

This is Certified that the project entitled **“**ToDevelop a chatbot using AWS Lex and Lambda**”** which is a experimental & Simulation work carried out by Pranathi, Anusha, Shiva Prasad Reddy, Mahalakshmi in partial fulfillment of the course requirements for the award of grades in the subject of **CLOUD AND SERVERLESS COMPUTING**, during the year **2024-2025**. The project has been approved as it satisfies the academic requirements.

**Ms.Saritha Dr. Arpita Gupta**

**Course Coordinator Head of the Department**

**CONTENTS**

1. Introduction

2. AWS Services Used as part of the project

3. Steps involved in solving project problem statement

4. Stepwise Screenshots with brief description

5. Learning Outcomes

6. Conclusion

7. References

**Chapter-1**

1. **INTRODUCTION**

*Mini Project Title*

Chatbots have become an essential tool for businesses and applications, enabling automated interactions with users. They improve customer engagement, streamline processes, and provide instant responses to queries. AWS provides a robust cloud-based solution to build chatbots using Amazon Lex, which integrates natural language understanding (NLU) and speech recognition.

Amazon Lex allows developers to create intelligent chatbots that understand user intent, extract meaningful information, and respond dynamically. It supports both text and voice inputs, making it a versatile tool for various applications, including customer support, information retrieval, and transaction processing.

This project focuses on developing a chatbot using AWS Lex and AWS Lambda in the AWS Console Free Tier account. AWS Lambda acts as the backend logic processor, executing necessary computations, retrieving data, and generating relevant responses. The chatbot will be designed to interact with users, process their queries, and provide appropriate responses dynamically.

By leveraging AWS services such as IAM, CloudWatch, and S3, developers can securely deploy and monitor the chatbot’s performance. This document outlines the AWS services used and the step-by-step approach to building a functional chatbot using AWS Lex and Lambda.

**Chapter - 2**

**2. AWS SERVICES USED AS PART OF THE PROJECT**

* **Amazon Lex:** A fully managed AI service designed for building conversational interfaces. It enables developers to create chatbots that understand natural language inputs and generate appropriate responses. Amazon Lex allows for defining intents (specific chatbot functionalities), slots (user-provided parameters), and utterances (sample phrases that trigger intents).
* **AWS Lambda:** A serverless compute service that executes backend logic for the chatbot. AWS Lambda runs code in response to Lex triggers and helps process user inputs, fetch relevant data, and generate responses dynamically. It eliminates the need for managing servers and ensures scalable execution.
* **Amazon CloudWatch**: A monitoring and logging service used to track chatbot interactions and troubleshoot issues. CloudWatch captures execution logs from AWS Lambda functions, enabling developers to diagnose errors and optimize performance.
* **Amazon S3**: A scalable storage service that can be used to store chatbot-related data, such as conversation logs, configuration files, or additional response templates.
* **AWS IAM**: The Identity and Access Management service controls access to AWS resources, ensuring that only authorized users and services can interact with Amazon Lex and AWS Lambda. Proper IAM roles and permissions are configured to enhance security.

**Chapter - 3 3. STEPS INVOLVED IN SOLVING PROJECT PROBLEM STATEMENT**

**Step 1: Setting Up Amazon Lex**

* Sign in to the AWS Management Console and navigate to Amazon Lex.
* Create a new Lex bot, specifying a name and selecting a template or starting from scratch.
* Define intents, which represent different functionalities of the chatbot (e.g., booking an appointment, answering FAQs).
* Set up slots to capture user input values, such as date, time, or location.
* Add sample utterances that users might say to trigger each intent.
* Configure response messages that Lex should return to users for different scenarios.

**Step 2: Configuring AWS Lambda for Fulfillment**

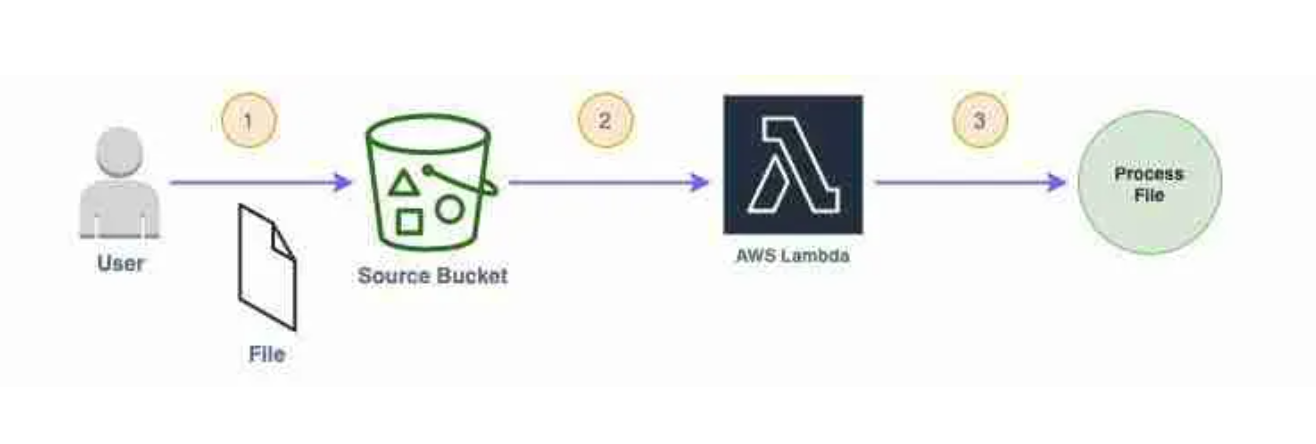
* Navigate to AWS Lambda in the AWS Console and create a new function.
* Choose "Author from scratch," select the runtime (Python or Node.js), and configure necessary permissions.
* Write a function that processes user input received from Lex and returns appropriate responses.
* Implement logic for handling different intents and retrieving necessary information.
* Deploy the Lambda function and copy the Amazon Resource Name (ARN) to integrate with Lex.

**Step 3: Integrating Lambda with Amazon Lex**

* Return to Amazon Lex and associate the Lambda function with the chatbot’s fulfilment options.
* Configure IAM permissions to allow Amazon Lex to invoke the Lambda function securely.
* Test integration by triggering the chatbot’s intents and validating Lambda responses.

**Step 4: Testing and Deploying the Chatbot**

* Use the Amazon Lex test console to simulate conversations and validate chatbot responses.
* Monitor chatbot interactions using Amazon CloudWatch Logs and debug issues if necessary.
* Deploy the chatbot to various platforms such as Amazon Connect, Slack, or embed it in a web interface.
* Continuously refine the chatbot by improving intent detection, refining responses, and handling edge cases.

****

**Chapter – 4 4. STEPWISE SCREENSHOTS WITH BRIEF DESCRIPTION**

**Chapter – 5**

**5. LEARNING OUTCOMES**

**Chapter – 6**

**6. CONCLUSION**

**Chapter - 7**

**7. REFERENCES**

AWS Documentation:

* Creating an Amazon Lex Bot: <Link>
* Integrating AWS Lambda with Amazon Lex: <Link>

GitHub Repositories:

* AI Chatbot for IoT Data: <Link>
* Lex Chatbot with Web UI: <Link>